Spring Boot

Introduction

Spring Boot is an open-source micro framework maintained by a company called Pivotal. It provides Java developers with a platform to get started with an auto configurable production-grade Spring application. With it, developers can get started quickly without losing time on preparing and configuring their Spring application.

Spring Boot is built on top of the Spring framework, and it comes with many dependencies that can be plugged into the Spring application. Some examples are Spring Kafka, Spring LDAP, Spring Web Services, and Spring Security. However, developers have to configure each building brick themselves using a lot of XML configuration files or annotations.

It generally functions by providing defaults for the codes and annotation configuration that will help you to instantly start any new Spring project in real-time. It also follows the 'Opinionated Defaults Configuration' strategy to eliminate boilerplate and other configurations designed to improve unit testing, development, and integration test procedures.

Advantages

These are some of the advantages of Spring Boot -

- Fast and easy development of Spring-based applications;
- No need for the deployment of war files;
- The ability to create standalone applications;
- Helping to directly embed Tomcat, Jetty, or Undertow into an application;
- No need for XML configuration;
- Reduced amounts of source code;
- Additional out-of-the-box functionality;
- Easy start;
- Simple setup and management;
- Large community and many training programs to facilitate the familiarization period.

With features like auto-configuration, Spring Boot saves you the hassle of coding and unnecessary configuration.

Spring Initializr

Spring Initializr is a **web-based tool** provided by the Pivotal Web Service. With the help of **Spring Initializr**, we can easily generate the structure of the **Spring Boot Project**. It offers extensible API for creating JVM-based projects.

It also provides various options for the project that are expressed in a metadata model. The metadata model allows us to configure the list of dependencies supported by JVM and platform versions, etc. It serves its metadata in a well-known that provides necessary assistance to third-party clients.

Spring Initializr Modules

Spring Initializr has the following module:

- **initializr-actuator:** It provides additional information and statistics on project generation. It is an optional module.
- **initializr-bom:** In this module, **BOM** stands for **Bill Of Materials**. In Spring Boot, BOM is a special kind of **POM** that is used to control the **versions** of a project's **dependencies**. It provides a central place to define and update those versions. It provides flexibility to add a dependency in our module without worrying about the versions.
  Outside the software world, the **BOM** is a list of parts, items, assemblies, and other materials required to create products. It explains **what, how,** and **where** to collect required materials.
- **initializr-docs:** It provides documentation.
- **initializr-generator:** It is a core project generation library.
- **initializr-generator-spring:**
- **initializr-generator-test:** It provides a test infrastructure for project generation.
- **initializr-metadata:** It provides metadata infrastructure for various aspects of the projects.
- **initializr-service-example:** It provides custom instances.
- **initializr-version-resolver:** It is an optional module to extract version numbers from an arbitrary POM.
- **initializr-web:** It provides web endpoints for third party clients.